Audio Events Detection in Public Transport Vehicle

Jean-Luc Rouas, Jérôme Louradour, Sébastien Ambellouis

▶ **To cite this version:**

Jean-Luc Rouas, Jérôme Louradour, Sébastien Ambellouis. Audio Events Detection in Public Transport Vehicle. 9th International IEEE Conference on Intelligent Transportation Systems (ITSC'2006), 2006, Toronto, Canada. hal-00664991

HAL Id: hal-00664991
https://hal.science/hal-00664991

Submitted on 31 Jan 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Audio Events Detection in Public Transport Vehicle

Jean-Luc Rouas, Jérôme Louradour, Sébastien Ambellouis

***Abstract*— This paper addresses the problem of automatic audio analysis for aided surveillance application in public transport. The aim of such application is to detect critical situations and to warn the control room. We propose a comparative study of two methods of modelisation/classification of acoustical segments. The problem is quite similar to the "'audio indexing"' framework, nevertheless the environment here is very noisy. We present two general frameworks based on Gaussian Model Mixture (GMM) and Support Vector Machine (SVM) to achieve shout detection in railway embedded environment.**

***Index Terms*— Audio processing, classification, transport, surveillance.**

I. Introduction

IMPROVING security and safety of public transport system is a major priority of operating companies which have deployed video surveillance thanks to CCTV systems. The installed systems are composed of ever-increasing number of cameras connected to a control room. The monitoring task requires too much workload for the operator to maintain a high level of attention and a short reaction time. Over the last decades, many researchers have been working on developing image processing tools to assist the operator by automatically detecting abnormal situations. In several projects, a set of tools, specifically dedicated to metro station environment, have been provided and evaluated in real situations [1], [2].

Nowadays, CCTV systems are deployed in embedded areas like train, bus or metro-vehicles. To reach the same objectives of security improvement and work flow reduction, video processing has to be adapted by taking into account the mobility constraints. A single visual analysis is not always sufficient to reliably understand passengers activity. It is often the case in overcrowded environments where occlusions appear. In this context, it is very difficult to isolate each passenger and to track its activity efficiently. Even if visual information can be extracted without too much difficulties, it can remain impossible to discriminate several activities when the behaviours have a quite similar visual description. When audio features are components of the behaviours model, sound can be a salient information to deal with ambiguities and to improve the detection/identification rate. Sound information become impossible to circumvent for an event that can't be modeled by video features and in area where video-surveillance can not be deployed.

This work was supported by the french EVAS and SAMSIT project.

Jean-Luc Rouas and Sébastien Ambellouis are with INRETS-LEOST, 20 rue Elisée Reclus, 59666 Villeneuve d'Ascq, FRANCE jean-luc.rouas@inrets.fr, sebastien.ambellouis@inrets.fr

Jérôme Louradour is with IRIT UMR 5505 CNRS, 31062 Toulouse, FRANCE louradou@irit.fr

Automated audio analysis is a challenging problem. A large amount of work have been carried out in speech recognition, in audio segmentation and classification and more recently, in audio source separation and localisation. In this project, we are interested in segmentation and classification of audio events. Recently, this research have been proposed in a medical telemonitoring application where the objective is to help patients at home or at hospitals when potentially dangerous situations appear [3]. The proposed system has been evaluated in a real environment and good results have been reached but in moderate noise conditions.

Before classifying audio events, the first step is to extract relevant events from the audio stream. This is achieved by an automatic audio event extraction algorithm based on an automatic segmentation algorithm and an activity detection. This algorithm is described in section II. In section III, after a brief description of speech characterisation features used, we focus on two kinds of classification techniques based on different paradigms: the Gaussian Mixture Model (GMM) and the Support Vector Machine (SVM) classifier. Experiments on the learning part of the corpus are described in section IV in order to assess the system's structure (front-end processing and classification tree). Cross-validation experiments are then achieved in section V to evaluate the different feature sets and to test the generalisation power of the models. A discussion is then proposed in section VI.

II. Segmentation in activity zones

The goal of the front-end processing is to extract relevant audio samples (activity segments) from the complete audio stream to reduce computing time and to improve global performance. It is based on 3 steps:

- An automatic audio segmentation, which splits an audio signal in several quasi-stationary consecutive zones,
- An activity detection algorithm, which aims at skipping silence and low-level noise zones, out of interest and
- A merging step, to gather successive activity segments.

A. Audio segmentation

The segmentation is issued from the "'Forward-Backward Divergence"' (DFB) algorithm [4], which is based on a statistical study of the signal in the temporal framework. This algorithm has been firstly applied to speech [4] and then successfully applied to speech/music segmentation [5] The audio signal is hypothetically described by a sequence of quasi-stationary segments. Each segment is characterised by a statistic model, the autoregressive Gaussian model:

$$\begin{cases} y_n = \sum a_i y_{n-i} + e_n \\ var(e_n) = \sigma_n^2 \end{cases} \quad (1)$$

where (y_n) is the speech signal and (e_n) is a Gaussian white noise.

The method consists in detecting the changes in the autoregressive models through the prediction errors computed on two analysis windows (figure 1). The distance between the two models is obtained by computing the mutual entropy of the two corresponding conditional laws.

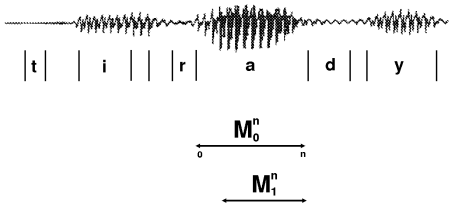

Fig. 1. Localisation of the estimation windows of models M_0^n and M_1^n at time n; time "0" corresponds to the last validated boundary. The french sentence is "il se garan**tira du**...".

The statistic is defined as a cumulative sum : $W_n = \sum_{k=1}^{n} w_k$. w_k, the mutual entropy between the two models in the Gaussian framework is $w_k = \frac{1}{2}\left\{2\frac{e_k^0 e_k^1}{\sigma_1^2} - [1+\frac{\sigma_0^2}{\sigma_1^2}]\frac{e_k^0}{\sigma_0^2} + [1-\frac{\sigma_0^2}{\sigma_1^2}]\right\}$. The prediction error for each model at instant k is $e_k^i = y_k - \sum_{j=1}^{p} a_j^i y_{k-j}, i = 0, 1$.

This method has been compared to numerous other segmentation methods [6]. It has given interesting results for automatic speech recognition: experiments have shown that the segments duration carries a relevant information [7].

The segmentation achieves an infra-phonemic segmentation where three kinds of segments can be identified:

- quasi-stationary segments, corresponding to the stable part of phonemes,
- transient segments,
- short segments (about 20 ms).

Their lengths vary between 20 and 100 ms for speech (figure 2).

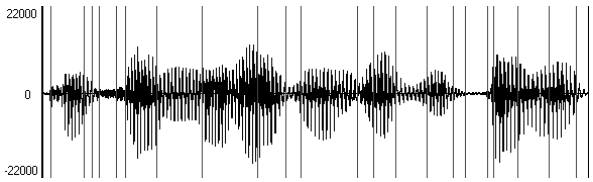

Fig. 2. Result of the segmentation on about 1 second of speech. The sentence is: "Confirmez le rendez-vous par écrit".

B. Activity segments detection and merging

The vocal activity detection is based on a first order statistic analysis of the temporal signal [8]. This algorithm has been developed by François Pellegrino, and previously integrated in an automatic language identification algorithm [9].

The activity detection algorithm detects the less intense segment of the excerpt (in terms of energy) and the other segments are classified as Silence or Activity according to an adaptive threshold. One can distinguish silences showing an absence of activity (long segments) and silences occurring during a sentence (short pauses, stops).

Time short segments are provided by the segmentation and the detection steps. To discard short segments (duration under $300ms$), we merge quasi-adjacent segments. The activity segments are quasi-adjacent if they are separated by a non activity segment which duration is under $300ms$. An example of a result obtained by this method is shown on figure 3.

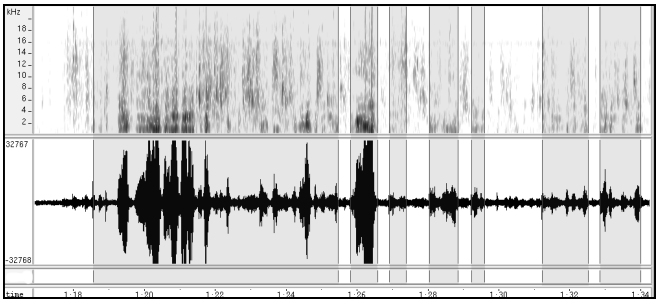

Fig. 3. Activity zones detected on an audio signal (in grey)

III. Modelling and classification framework

We have used two methods to model the audio data: the classical Gaussian Mixture Models (GMM) and the Support Vector Machine (SVM) technique. Both methods are applied to acoustic parameters extracted from the audio signal.

A. Feature extraction

We have studied the impact of using different features that are widely used in speech processing: the Mel Frequency Cepstral Coefficients (MFCC), the Linear Prediction Coefficients (LPC) and the Perceptual Lineal Prediction Coefficients (PLP). The MFCC and the PLP are computed after a transformation of the signal in the spectral domain. LPC is based on a predictive analysis assuming that a speech sample at a current time can be approximated as a linear combination of past speech samples.

We have completed each coefficient set by their temporal derivatives and their accelerations and we have added the energy coefficient with its derivative and acceleration. Finally, the coefficients set we used in the experiments are:

1) 12 MFCC, energy, Δ, $\Delta\Delta$ (39 coefficients),
2) 20 MFCC, energy, Δ, $\Delta\Delta$ (63 coefficients),
3) 12 LPC, energy, Δ, $\Delta\Delta$ (39 coefficients),
4) 12 PLP, energy, Δ, $\Delta\Delta$ (39 coefficients).

B. GMM

This method supposes that the different classes which are represented in the feature space can be modeled with a weighted sum of Gaussian distributions. The parameters of the Gaussian mixture are estimated using the EM

(Expectation-Maximisation) algorithm initialised with the LBG algorithm [10].

Let $X = \{x_1, x_2, .., x_N\}$ be the training set and $\Pi = \{(\alpha_i, \mu_i, \sigma_i), 1 \leq i \leq Q\}$ the parameter set that defines a mixture of Q p-dimensional Gaussian pdfs. The model that maximises the overall likelihood of the data is given by:

$$\Pi^* = \arg\max_{\pi} \prod_{i=1}^{N} \left\{ \sum_{k=1}^{Q} \frac{\alpha_k}{(2\pi)^{p/2}\sqrt{|\Sigma_k|}} \exp\left[-\frac{1}{2}(x_i - \mu_k)^t \Sigma_k^{-1} (x_i - \mu_k) \right] \right\} \quad (2)$$

where α_k is the mixing weight of the k^{th} Gaussian term. The maximum likelihood parameters Π^* are obtained using the EM algorithm. This algorithm presupposes that the number of components Q and the initial values are given for each Gaussian pdf. Since these values greatly affect the performances of the EM algorithm, a vector quantization (VQ) is applied to the training corpus to optimize them.

During the identification phase, all the activity segments detected in the test utterance are gathered and parameterised. The likelihood of this set of segments $Y = \{y_1, y_2, ..., y_N\}$ according to each model (denoted C_i) is given by $P(Y|C_i) = \prod_{j=1}^{N} P(y_j|C_i)$, where $P(y_j|C_i)$ denotes the likelihood of each segment. Under the Winner Takes All (WTA) assumption [11], $P(y_j|C_i)$ is approximated by:

$$P(y_j|C_i) = \max_{1 \leq k \leq Q_i} \left\{ \frac{\alpha_k^i}{(2\pi)^{p/2}\sqrt{|\Sigma_k^i|}} \exp\left[-\frac{1}{2}(y_j - \mu_i^k)^t \Sigma_k^{-1} (y_j - \mu_k^i) \right] \right\} \quad (3)$$

C. SVM

The GMM is capturing the feature distributions of several classes and perform classification using a bayesian criterion decision. On the contrary, the SVM technique directly focus on modelling a discriminative function to separate the classes. This function is a linear combination of several *kernel* functions k estimated on some training data. For a binary classification problem, involving training data $\left(x_i^+\right)$ (resp. $\left(x_i^-\right)$) with labels $l_i = +1$ (resp. -1), it can be written $f(y) = \sum \alpha_i^+ k(y, x_i^+) - \sum \alpha_i^- k(y, x_i^-) + b$.

The weights α_i are positive, b is a threshold such that $f(y) > 0$ means that we should decide to affect the label $+1$ to y, and $k(.,.)$ can be seen as a generalised dot product. The training process is an optimisation problem, in which the regularised cost to minimise can be seen as a weighted sum of the empirical risk (overall difference with target values $l_i = +1/-1$) on the training corpus, and a complexity term to control capacity and prevent overfitting:
$C \frac{1}{N} \sum_{i=1}^{N} \mid l_i - f(x_i) \mid_+ + T(\alpha)$.

The trade-off parameter C plays the same role as the number of Gaussian mixtures in the GMM approach. Because of discontinuity of the first derivatives of the regularised cost, involving the loss function $|z|_+ = \max(0, |z|)$, many components of the optimal weight vector α are zero, i.e. a sparse solution is obtained, leading to a quite fast-scoring procedure. Non-zero α_i correspond to *support vectors*, which define, with the function k, the complexity of the decision frontier ($f(y) = 0$). If the *gram matrix* $\mathbf{K}$, composed of kernel evaluations between every pair of training data ($\mathbf{K}_{i,j)} = k(x_i, x_j)$), is definite positive, then it is possible to find the optimal solution of the optimisation problem. Otherwise, the training algorithm is not guaranteed to converge.

In the case of audio data classification, each acoustic vector taken individually contains little discriminative information, and discriminative techniques applied at the vector level suffer from the noise. Hence the interest of processing at a higher scale: the sequence level. As the optimal way to combine SVM outputs was not yet found (contrary to GMM that offers a probabilistic framework which allows to process sequences in a natural way), we investigate here SVM with sequence kernels.

An efficient technique which has shown good performance in speaker verification (a typical problem of audio sequence classification), is the SVM using a Generalised Linear Discriminant Sequence (GLDS) Kernel [12]. The kernel computation between two sequences amounts (with a few practical approximations) to learn a vector-level polynomial classifier on one sequence and to test on the other.

The GLDS kernel computation involves a polynomial expansion ϕ_p, composed of all monomials between every possible combination of vector components up to a given degree p. For example, if $p = 2$ and $\mathrm{x} = [x_1, x_2]^\top$ is a 2-dimensional input vector, $\phi_p(\mathrm{x}) = [x_1, x_2, x_1^2, x_1 x_2, x_2^2]^\top$. The GLDS kernel between two sequences of vectors $X = \{\mathrm{x}_t\}_{t=1...T_X}$ and $Y = \{\mathrm{y}_t\}_{t=1...T_Y}$ is given as a rescaled dot product between average expansions :

$$k_{GLDS}(X, Y) = \frac{1}{T_X} \sum_{t=1}^{T_X} \phi_p(\mathrm{x}_t)^\top \; \mathbf{M}_p^{-1} \; \frac{1}{T_Y} \sum_{s=1}^{T_Y} \phi_p(\mathrm{y}_s) \quad (4)$$

where $\mathbf{M}_p$ is the second moment matrix of polynomial expansions ϕ_p estimated on some background population, or its diagonal approximation for more efficiency. In our experiments we use this approximation and $p = 3$, with a view to having a first idea of the performance of a SVM classifier.

D. Classification Framework

The hierarchical tree is build according to figure 4. Several options are studied regarding the topology of the tree.

1) During the first step, a background noise/short term noises detection is achieved. The background noise model is composed with every parts of the signal where the noise is present, including speech and shout parts. The short term noise model is learnt on particular noise segments (opening of doors, bags hitting the ground ...)
2) During the second step, a speech/non speech classification is done. For this step, we assume human shouts as part of the speech model.

3) Finally, a shout/non shout decision is made.

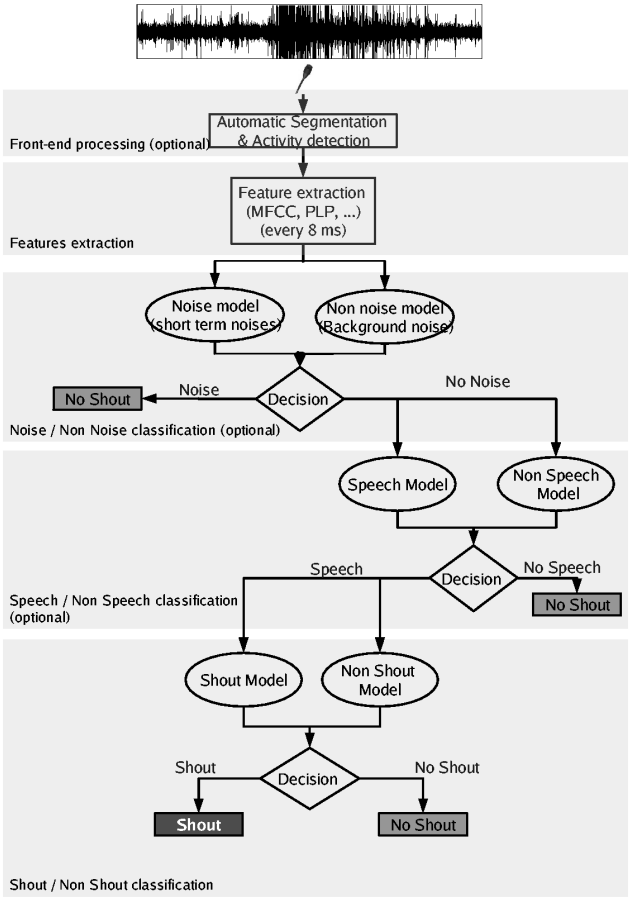

Fig. 4. Complete hierarchical tree used for classification

IV. Tuning experiments

We ran the modelisation/classification process on the learning part of the corpus in order to confirm or infirm the relevance of the front-end processing and the classification tree. Those experiments allow to see which are the best technical choices. For each experiment, a confusion matrix is drawn. We focus on shout detection rate and false detection rate (Shouts identified while they should not be). The system aims at helping surveillance operators by providing alarms when shouts are detected. If the system gives too much false alarms, it will not be used anymore.

A. Description of the Corpus

The audio data of the corpus were recorded by ourselves in a regional train in the SAMSIT project context. Scenes were recorded using simultaneously 4 microphones. Actors were asked to play scenarios representative of the public french train operator's needs:

- Scenario 1: fight scene involving two people or more,
- Scenario 2: fight scene involving two men and a women,
- Scenario 3: violent robbery scene (two guys attack one person),
- Scenario 4: bag or mobile snatching (one lady).

Each scenario is played several times. Moreover, for each kind of scenario, the actors played a scene which is not a critical situation but has similar acoustic properties. The scene is called the "'normal"' situation. The total corpus duration is approximatively 2500 seconds, and the total duration of shouts is approximatively $140s$. All files of the corpus have been labelled. The hand labelled shouts have a mean duration of $2.85s$.

B. Relevance of the front-end processing

We have shown the relevance of the front-end processing (segmentation in activity zones) in the GMM context. We achieved experiments using 12 MFCC with energy, derivatives and accelerations.

The table I shows the best results obtained without front-end processing. Feature vectors are computed on all the signal, each $8ms$ and a smoothing is applied to provide results each $200ms$.

TABLE I

Results on the learning part of the corpus, without front-end processing and without hierarchical tree

Result → ↓ Expected	Non Shouts	Shouts
Non Shouts (2402 s.)	85.0% (2043 s.)	15.0% (360 s.)
Shouts (138 s.)	24.1% (35 s.)	75.9% (109 s.)

As a comparison, experiments are also made using the same system, including the front-end processing. In this context, the features and a decision are computed on each activity zone. The table II shows the best results we obtained.

TABLE II

Results on the learning part of the corpus, with front-end processing and without hierarchical classification tree

Result → ↓ Expected	Non Shouts	Shouts
Non Shouts (2402 s.)	97.0% (2330 s.)	3.0% (73 s)
Shouts (138 s)	24.2%(33 s)	75.8% (105 s)

Considering shout detection, results are very similar whether the front-end processing is used or not (about 75% of correct detections). More importantly, the false alarms rate decrease significantly if we use the front-end processing (only 73 s of misclassified shouts against 360 s). These experiments show the relevance of the front-end processing, which does not influence the shout detection performance while decreasing the number of false alarms.

C. Relevance of the classification tree

These experiments aim at verifying whether the use of the classification tree can improve the performance of the system. In both cases, system settings are the same, except for the use of the classification tree. The front-end processing is applied. The features used are 12 MFCC with energy, Δ and $\Delta\Delta$. Models used are in both cases Gaussian Mixture

Models. Results obtained without using the classification tree are displayed in table II.

As we said previously, the classification tree is composed of a background noise or other noise decision, and a speech / non speech decision. For the noise classification, other noise models are trained using short term noise excerpts of the corpus (i.e. door noises, etc.). The background noise model is trained using all the remaining parts of the corpus (including speech and shouts). The speech model is learnt using any part of the corpus containing speech, including shouts. The non speech model is learnt using every non speech parts of the corpus.

Results obtained with the classification tree are displayed in table III.

TABLE III

RESULTS ON THE LEARNING PART OF THE CORPUS, WITH FONT-END PROCESSING AND WITH COMPLETE HIERARCHICAL TREE

Result → ↓ Expected	Non Shouts	Shouts
Non Shouts (2402 s.)	98.3 % (2363 s)	1.6 % (39 s)
Shouts (138 s)	25.6% (35 s)	74.3% (103 s)

Considering shout detection, results are very similar whether the classification tree is used or not (about 75% of correct detections), while being a little under performance without using the classification tree ($103s$ versus $105s$ correct). The false alarm rate decrease if we use the classification tree (only $39s$ of misclassified shouts against $73s$).

V. CROSS-VALIDATION EXPERIMENTS

Cross-validation aims at estimating how well the model we have learned from some training data is going to perform on future unknown data. We have choosen the Leave-one-out Method. This method involves in three steps. Firstly, the model is trained on all the training data except for one. Secondly, the learned model is evaluated on the remaining data. Both steps are repeated such that each data is used once as the validation data. The evaluation process we achieved focuses not only on the good or bad detection of events, but also on the precision on the time scale of the detection. The results are then expressed as correctly identified durations (for shouts and non shouts) and misidentified durations. We present only the shout detection rate and the shout false alarm rate.

This procedure is repeated for all the files of the corpus and for both classification algorithms (GMM and SVM). In the GMM context, we test different number of Gaussian laws. From the SVM point of view, the C parameter is varying within the interval $[0.01\ 10]$. To save space and to ease reading, results are displayed graphically. The correct identified shouts duration and the misidentified shouts duration are respectively in white and grey bars. The shout false alarm duration is the black curve.

Results obtained with Gaussian Mixture Models are displayed in figure 5.

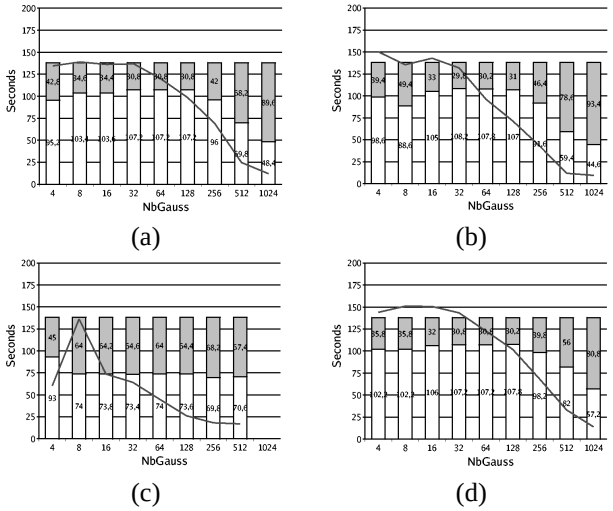

Fig. 5. Results obtained in cross validation experiments with Gaussian Mixture Models. Correct identifications correspond to the white parts of bars. Wrong classifications are represented by the grey part of bars. False alarms are displayed as a curve. (a) 12 MFCC + E + Δ + ΔΔ (b) 20 MFCC + E + Δ + ΔΔ (c) 12 LPC + E + Δ + ΔΔ (d) 12 PLP+ E + Δ + ΔΔ

These graphics show that the least false error rate is obtained using 1024 Gaussian laws in the mixture and for the feature sets (b) and (d). The duration of the false alarms, respectively $9.4s$. and $14s$, is relatively low compared to the total duration of the corpus ($\approx 2540s$) and the total duration of shouts to identify ($\approx 140s$).

The table IV shows results in a different way for these two features sets. On this table, one can see that while there are no shouts to be detected ("'Normal"' cases), the algorithm does produce very few false alarms (i.e. only $1.3s$ shout identified for the "'Normal"' condition of the second scenario).

Shout detection rate does not seem very good. However, even if all shouts are not always identified on the "'Scene"' cases, the GMM method with PLP features set perform well in terms of number of "'identified shouts"'. A critical situation can be composed of several shouts and the detection of a part of the shouts can be sufficient to detect a critical situation and to set off an alarm.

TABLE IV

RESULTS FOR THE GAUSSIAN MIXTURE MODEL CLASSIFIER, 20 MFCC AND 12 PLP FEATURE SETS. RESULTS ARE DISPLAYED AS NUMBER AND DURATION OF SHOUTS TO BE IDENTIFIED FOR EACH SCENARIO.

Scenario	Scene	Hand Labels	GMM (12 PLP)	GMM (20 MFCC)
Scenario 1	Normal	0 (0 s.)	3 (1.2 s.)	2 (0.8 s.)
	Scene	5 (28.4 s.)	4 (10.4 s.)	13 (27.8 s.)
Scenario 2	Normal	0 (0 s.)	1 (1.3 s.)	1 (1.3 s.)
	Scene	17 (57.4 s.)	17 (34.4 s.)	6 (8.8 s.)
Scenario 3	Normal	0 (0 s.)	1 (0.5 s.)	0 (0 s.)
	Scene	17 (43 s.)	14 (24.3 s.)	8 (16.9 s.)
Scenario 4	Scene	9 (9.2 s.)	9 (9.5 s.)	5 (4.0 s.)

The same experiments have been done with the SVM classifier approach. Results are summarised in figure 6.

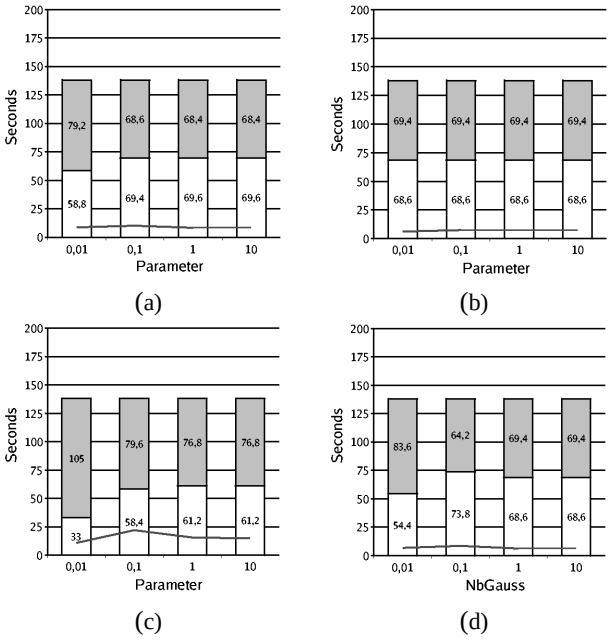

Fig. 6. Results obtained in cross validation experiments with Support Vector Machines. Correct identifications correspond to white parts of bars. Wrong classifications are represented by grey part of bars. False alarms are displayed as a curve.(a) 12 MFCC + E + Δ + ΔΔ (b) 20 MFCC + E + Δ + ΔΔ (c) 12 LPC + E + Δ + ΔΔ (d) 12 PLP+ E + Δ + ΔΔ

The graphics show that the SVM classification framework results in less false alarms than GMM. The most effective feature sets are still 20 MFCC and PLP, with total false alarms duration respectively of $6.2s$ and $7.4s$. Meanwhile performance in terms of duration of identified shouts seems better than for the GMM case.

We have also further investigated results for those two features sets using the SVM classifier. Results are presented in table V. This table allows us to verify that the SVM performs well in terms of false alarms, except in the "'normal"' scene of the Scenario 2, for which one or two shouts are detected. As for the GMM, the PLP features set provides slightly better performance than the 20 MFCC features set.

TABLE V

RESULTS FOR THE SUPPORT VECTOR MACHINE CLASSIFIER, 20 MFCC AND 12 PLP FEATURE SETS. RESULTS ARE DISPLAYED AS NUMBER AND DURATION OF SHOUTS TO BE IDENTIFIED FOR EACH SCENARIO.

Scenario	Scene	Hand Labels	SVM (12 PLP)	SVM (20 MFCC)
Scenario 1	Normal Scene	0 (0 s.) 5 (28.4 s.)	0 (0 s.) 5 (18.7 s.)	0 (0 s.) 4 (14.8 s.)
Scenario 2	Normal Scene	0 (0 s.) 17 (57.4 s.)	1 (1.3 s.) 11 (29.1 s.)	2 (2.4 s.) 10 (25.5 s.)
Scenario 3	Normal Scene	0 (0 s.) 17 (43 s.)	0 (0 s.) 11 (23.6 s.)	0 (0 s.) 14 (28.3 s.)
Scenario 4	Scene	9 (9.2 s.)	8 (6.8 s.)	6 (6.0 s.)

When comparing GMM and SVM using the PLP features set, we can see that the SVM approach generates less false alarms than the GMM classifier: one false alarm for a duration of 1.3 seconds versus 5 false alarms for a duration of 2.98 seconds. But the shout identification performance is worst for the SVM classifier than for the GMM approach.

VI. CONCLUSIONS AND PERSPECTIVES

We have presented and compared two modelling/classification methods to detect shout events in a public transport vehicle. Both methods have been evaluated on real life railway environment. Vehicle environment is quite noisy and both approaches achieved promising performance.

We have shown that SVM method generates a weak false alarms rate and that GMM approach has better identification rate. In a surveillance application, it is important not to generate too much false alarms. In the case of shout detection, missing some shouts may not be critical if we can detect a significant number of shouts to raise the alarm. Thus, a compromise has to be done to choose one between these GMM and SVM classifiers. As a consequence the PLP feature set combined with the SVM classifier should be the best choice for our application.

We are working to adapt both methods in an urban bus context. This environment is more constrained because of the vehicle vibrations, the motor noise and the sounds outside the bus.

REFERENCES

[1] L. Khoudour, D. Aubert, J.-L. Bruyelle, T. Leclercq, and A. Flancquart, *A distributed multi-sensor surveillance system for public transport applications*. IEE, to appear, ch. 7.

[2] F. Cupillard, F. Bremond, and T. M., *The Kluwer International Series in Computer Vision and Distributed Processing Video-Based Surveillance Systems*. Kluwer Academic Publishers, 2002, ch. Tracking Group of People for Video Surveillance.

[3] M. Vacher, D. Istrate, and J.-F. Serignat, "Sound detection and classification through transient models using wavelet coefficient trees," in *European Signal Processing Conference*, Vienna, Austria, september 2004, pp. 1171 – 1174.

[4] R. André-Obrecht, "A new statistical approach for automatic speech segmentation," *IEEE Transanctions on Acoustics, Speech and Signal Processing*, vol. 36, no. 1, pp. 29–40, 1988.

[5] J. Pinquier, J.-L. Rouas, and R. André-Obrecht, "Fusion de paramètres pour une classification automatique parole/musique robuste ," in *Technique et science informatiques (TSI) : Fusion numérique/symbolique* . 8, quai du marche neuf, F-75004 Paris: Hermès, 2003, vol. 22, pp. 831–852.

[6] R. André-Obrecht, "Segmentation et parole ?" Habilitation à diriger des recherches, Université de Rennes - IRISA, Rennes, June 1993.

[7] R. André-Obrecht and B. Jacob, "Direct identification vs. correlated models to process acoustic and articulatory informations in automatic speech recognition," in *International Conference on Acoustics, Speech and Signal Processing*. Munich: IEEE, 1997, pp. 989–992.

[8] F. Pellegrino and R. André-Obrecht, "Vocalic system modeling : A vq approach," in *IEEE Digital Signal Processing*, Santorini, July 1997, pp. 427–430.

[9] J.-L. Rouas, J. Farinas, F. Pellegrino, and R. André-Obrecht, "Rhythmic unit extraction and modelling for automatic language identification," *Speech Communication*, vol. 47, no. 4, pp. 436–456, 2005.

[10] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, 1980.

[11] S. Nowlan, "Soft competitive adaptation: Neural network learning algorithm based on fitting statistical mixtures," Ph.D. dissertation, School of Computer Science, Carnegie Mellon University, 1991.

[12] W. Campbell, J. Campbell, D. Reynolds, E. Singer, and P. Torres-Carrasquillo, "Support vector machines for speaker and language recognition," *Computer Speech and Language*, vol. 20, no. 2-3, pp. 210–229, April-July 2006.